

## **Avaliativa: Lista Dinâmica/Encadeada**

**Objetivo:** Capacitar ao aluno desenvolver as operações básicas de listas utilizando a implementação dinâmica/encadeada simples e com nó descritor (nó cabeçalho).

### **Exercícios:**

1. Implementar, utilizando a alocação dinâmica e o acesso **encadeado simples** (SEM qualquer técnica de encadeamento), o TAD lista linear não ordenada de caracteres. Além das operações básicas (*criar\_lista*, *lista\_vazia*, *insere\_elem* e *remove\_elem*), o TAD também deve contemplar a operação *get\_elem\_pos*, que retorna o conteúdo do elemento na posição indicada, caso ela exista; e *apaga\_lista*, que elimina uma instância da lista. Por fim, também desenvolva um programa aplicativo que permita ao usuário criar, apagar e imprimir uma lista, bem como inserir e remover elementos nessa lista. Destaca-se que a criação da instância da lista só pode ser realizada uma única vez, caso ela ainda não exista. Por outro lado, as demais opções só devem ser executadas se houver a instância da lista. A lógica dessas consistências deve ser implementada no programa aplicativo.

**OBS:** na implementação dinâmica/encadeada simples, *esvaziar* e *apagar* uma lista envolvem os mesmos passos: liberar a memória alocada para cada elemento da lista (um a um) e retorná-la para o estado de vazia. Portanto, basta implementar uma delas. Vale destacar que, **NÃO** deve ser usada a operação *remove\_elem para apagar os nós da lista em outras operações (como no *apaga\_lista*), pois provocaria percorrimientos desnecessários, reduzindo a eficiência dessas operações. Além disso, a lógica de impressão da lista deve ser implementada no arquivo do aplicativo (TAD não deve ter interface com o usuário), por meio da operação *get\_elem\_pos*.*

2. Implementar, utilizando a alocação dinâmica, o acesso encadeado **COM nó cabeçalho** (nó descritor), o TAD lista linear ordenada de termos de um polinômio, sendo que cada termo é formado pelo seu coeficiente e expoente (ex:  $3x^2$ ). Nessa implementação a lista segue uma **ordenação crescente dos expoentes** e a estrutura do termo também deve estar encapsulada no TAD. Além das operações básicas (*criar\_lista*, *lista\_vazia*, *insere\_ord* e *remove\_ord*), o TAD também deve contemplar as operações: *get\_elem\_pos*; *apaga\_lista*; e *esvazia\_lista*. Por fim, também desenvolva um programa aplicativo que permita ao usuário criar, esvaziar, imprimir e apagar uma lista, bem como inserir e remover elementos nessa lista.

**OBS:** na implementação dinâmica/encadeada COM nó cabeçalho, *esvaziar* e *apagar* uma lista são diferentes. Ao esvaziar, apaga-se apenas os nós correspondentes aos elementos da lista, mantendo o nó cabeçalho, e retorna a lista para o estado de vazia. Ao apagar, apaga-se todos os nós da lista, inclusive o nó cabeçalho, e faz a lista apontar para NULL. Vale destacar que, **NÃO** deve ser usada a operação *remove\_ord*, pois provocaria percorrimientos extras desnecessários, reduzindo a eficiência da operação.